Algorithms-1 - CS21003
(Class Test I)
Date: 10 – October – 2020

Maximum marks: 25 Duration: 1 hour

File naming convention: e.g., 18CS3004_G3_CT1.pdf (or any other extension).

In case of multiple files, use _1, _2 etc at the end.

Submission is via Moodle only. Email submissions will NOT be accepted. Please manage your time well keeping in mind that Internet and power disruptions are a new normal!

No clarifications from the TAs today. You can make any assumption as long as it is rational and you clearly state the same while solving the problem.

Plagiarism, in any form (including Internet source) will be severely penalized.

Question 1

There are two BSTs i.e., T_1 and T_2. The inorder listing of vertices of T_1 is $\{5, 9, 12, 16, 19\}$. The inorder listing of vertices of T_2 is $\{1, 2, 3, 4, 7, 11\}$. Construct an *AVL Tree* T_b consisting of the union of the vertices from the two BSTs. Just draw the final AVL Tree T_b (you do not need to write an algorithm!). Also, print the post-order traversal listing of T_b.

[5 marks]

Question 2

Let $G = (V, E)$ be an undirected graph represented as an adjacency list. A subset of edges F is called a feedback edge set if F contains at least one edge of every cycle of G. The size of a feedback edge set is the number of edges in it. A minimum size feedback edge set is a feedback edge set with minimum size among all possible feedback edge sets of G. Design an $O(|V| + |E|)$ time algorithm to find a minimum size feedback edge set of G.
Use C/C++ style pseudocode for your algorithm. Please note that you can use any traversal function already covered in the class by its prototype, and do not need to write the pseudocode again.

[8 marks]

Question 3

Given different models for each garment (e.g., 3 models of shirts, 2 models of belts, 4 models of shoes, ...), *we need to buy one model of each garment*. As the budget is limited, we cannot spend more money than the budget, but we want to spend the maximum possible. But it is also possible that we cannot buy one model of each garment due to that small amount of budget. In that case, we will not be able to buy anything.
The input consist of two integers $1 \leq M \leq 200$ and $1 \leq C \leq 20$, where M is the budget and C is the types of garments from which you have to buy one model each. Then in the input, there are information of the C garments. For a `garment_id` $\in [0...C-1]$, we know an integer $1 \leq K \leq 20$ which indicates the number of different models for that `garment_id`, followed by K integers indicating the price of each model $\in [1...K]$ of that `garment_id`.
The output should consist of one integer that indicates the maximum amount of money you will be able to spend to buy one element of each garment without exceeding the initial amount of money. If there is no solution, print "no solution".
For example, if the input is like this (test case A):
$M = 20, C = 3$
3 models of `garment_id` 0: 6 4 $\underline{8}$ // see that the prices are not sorted in input
2 models of `garment_id` 1: 5 $\underline{10}$
4 models of `garment_id` 2: $\underline{1}$ 5 3 5

Then the answer is 19, which may come from buying the **underlined** items (8+10+1). Note that this solution is not unique, as we also have (6+10+3) and (4+10+5).

However, if the input is like this (test case B):

$M = 9$ (very limited budget), $C = 3$

3 models of `garment_id` 0: 6 4 8

2 models of `garment_id` 1: 5 10

4 models of `garment_id` 2: 1 5 3 5

Then the answer is "no solution" as buying all the cheapest models $(4+5+1) = 10$ is still $> M$.

Use dynamic programming (memoization not allowed) to solve this problem. Use C/C++ style pseudocode to only show the DP formulation part along with the base cases (if any).

[12 marks]